memcpy_s() and memmove_s()

Daniel Plakosh, Software Engineering Institute [vita[1]]

2005-09-27

Substituting the `memcpy_s()` and `memmove_s()` functions for the `memcpy()` and `memmove()` functions can help guard against software vulnerabilities.

Development Context

Copying characters from one memory location to another.

Technology Context

C++, C, UNIX, Win32

Attacks

Attacker executes arbitrary code on machine with permissions of compromised process or changes the behavior of the program.

Risk

The `memcpy()` and `memmove()` functions are a source of buffer overflow vulnerabilities.

Description

Substituting the `memcpy_s()` and `memmove_s()` functions for the `memcpy()` and `memmove()` functions can help guard against software vulnerabilities. The `memcpy_s()` and `memmove_s()` functions defined in ISO/IEC WDTR 24731 are similar to the corresponding `memcpy()` and `memmove()` functions but provide some additional safeguards. These functions have an additional argument that specifies the maximum size of the destination, and they also include a return value that indicates whether the operation was successful. A return value of zero indicates that the operation succeeded. A non-zero return value indicates that the operation failed because it was diagnosed to have an undefined behavior due to an invalid input argument.

The `memcpy_s()` and `memmove_s()` functions will be diagnosed to have an undefined behavior if either the source or destination pointer is null, if the specified number of characters to copy or move is greater than the maximum size of the destination buffer, or the number of characters to copy or move or the maximum size of the destination buffer is greater than `RSIZE_MAX` .[13] Additionally, the

1. daisy:268 (Plakosh, Daniel)

13. The `RSIZE_MAX` is used to limit the size of objects passed to functions that have parameters of type `rsize_t` . Extremely large object sizes are frequently a sign that an object's size was calculated incorrectly. For example, negative numbers appear as very large positive numbers when converted to an unsigned type like `size_t`. Also, some implementations do not support objects as large as the maximum value that can be represented by type `size_t` . As a result, it is sometimes beneficial to restrict the range of object sizes to detect potential vulnerabilities.

`memcpy_s()` function will be diagnosed to have an undefined behavior if the memory regions of the objects overlap.

If the operation is diagnosed to have an undefined behavior, zeros will be stored in the first characters of the destination if the destination pointer is not equal to null and the size of the destination buffer is less than or equal to `RSIZE_MAX` .

The `memcpy_s()` function has better performance than the `memmove_s()` but has additional risks. There is no security related reason to prefer `memcpy_s()` to `memmove_s()`.

The `memcpy_s()` and `memmove_s()` functions are used to copy characters from one memory location to another. The `wmemcpy_s()` and `wmemmove_s()` functions are used to copy wide characters.

References

[ISO/IEC 99]	ISO/IEC. *ISO/IEC 9899 Second edition 1999-12-01 Programming languages — C.* International Organization for Standardization, 1999.
[ISO/IEC 04]	ISO/IEC. *ISO/IEC WDTR 24731 Specification for Secure C Library Functions*. International Organization for Standardization, 2004.

Pearson Education, Inc. Copyright

Fields

Name	Value
Copyright Holder	Pearson Education

Fields

Name	Value
is-content-area-overview	false
Content Areas	Knowledge/Coding Practices
SDLC Relevance	Implementation
Workflow State	Publishable